

Deriving the Probabilistic Capacity of General Run-Length Sets Using Generating Functions

G. Böcherer, R. Mathar
Institute for Theoretical Information Technology
RWTH Aachen University
52056 Aachen, Germany
Email: {boecherer,mathar}@ti.rwth-aachen.de

V.C. da Rocha Jr., C. Pimentel
Communications Research Group - CODEC
Department of Electronics and Systems, P.O. Box 7800
Federal University of Pernambuco
50711-970 Recife PE, Brazil
E-mail: {vcr,cecilio}@ufpe.br

Abstract—In *Reliable Communication in the Absence of a Common Clock* (Yeung *et al.*, 2009), the authors introduce general run-length sets, which form a class of constrained systems that permit run-lengths from a countably infinite set. For a particular definition of probabilistic capacity, they show that probabilistic capacity is equal to combinatorial capacity. In the present work, it is shown that the same result also holds for Shannon’s original definition of probabilistic capacity. The derivation presented here is based on generating functions of constrained systems as developed in *On the Capacity of Constrained Systems* (Böcherer *et al.*, 2010) and provides a unified information-theoretic treatment of general run-length sets.

I. INTRODUCTION

A constrained system allows the noiseless transmission of input sequences of weighted symbols that fulfill certain constraints on the symbol constellations. The weight of a symbol can have different practical meanings, e.g., in the context of magnetic recording systems, “weight” usually refers to “tape-length”; other meanings like “time” or “energy” are possible, depending on the modeled system.

For the design of encoders for such systems, it is of interest to determine the maximum entropy rate at which a random process can generate strings that fulfill the constraints. This rate is called the *probabilistic capacity* and a process that reaches this rate is called *maxentropic*. The probabilistic capacity is equal to the combinatorial capacity in the case that the constraints form a regular language. This was originally shown in [1]. In [2], the authors show this property for a slightly generalized setup, since they allow non-integer valued symbol weights, as long as the set of weights is not too dense. The precise definition of “not too dense” is stated in Section II.

Recently, in [3] the authors introduced the continuous time asynchronous channel as a model for time jitter in a communication system with no common clock between the transmitter and the receiver. As a constrained system, such a channel is defined by an in general countably infinite set of distinguishable run-lengths \mathcal{W} and a finite set of labels \mathcal{L} . A run is a substring during which the label does not change. The channel then allows the noiseless transmission

of strings where each run has a length in \mathcal{W} and a label in \mathcal{L} and where in addition two subsequent runs always have different labels. We will refer to such channels by *general run-length sets*. Formally, general run-length sets are different from constrained systems that have been considered so far in two ways: first, the set of strings that are accepted do in general not form a regular language, and second, the set of distinct string-lengths does in general not fulfill the “not too dense” restriction. As a result, general run-length sets cannot be treated using the framework of finite state machines [2].

In [4], we presented a new framework for constrained systems based on generating functions. Differing from [2], this framework is neither restricted to regular constraints nor does it impose the “not too dense” restriction to the set of string-lengths. The key result of [4] is that the entropy rate of input processes is in general upper-bounded by the combinatorial capacity of the system. This result allows us to derive the equality of probabilistic and combinatorial capacity for general run-length sets in a unified manner: we first derive the generating function of general run-length sets, we then use the generating function to calculate the combinatorial capacity, and we finally define an input process whose entropy rate is equal to the combinatorial capacity and thus achieves the maximum. In exactly the same way we derived in [4] as an illustrating example the equality of probabilistic and combinatorial capacity for (j, k) constraints.

For a particular definition of probabilistic capacity, the equality of probabilistic and combinatorial capacity of general run-length sets was derived in [3]. The definition of probabilistic capacity in [3] is based on a notion of entropy rate that is different from Shannon’s in [1, Appendix 4]. Because of the different definitions of probabilistic capacity, the derivations presented in our work strongly differ from the derivations in [3]. We discuss the relation of our results to the results in [3] in Subsection VI-D.

The remainder of this work is organized as follows. In Section II, we state the results from [4] that we need in this work. In Section III, we formally define general run-length sets. In Section IV, V, and VI, we derive for general run-length sets the generating function, the combinatorial capacity, and the probabilistic capacity, respectively.

This work has been supported by the UMIC Research Centre, RWTH Aachen University. This work has also been supported by the Brazilian National Council for Scientific and Technological Development (CNPq).

II. CONSTRAINED SYSTEMS, GENERATING FUNCTIONS, AND CAPACITY

In this section, we shortly provide the main definitions and results from our work [4]. These form a mathematical framework based on generating functions for the information theoretic treatment of constrained systems. Within this framework, we will derive the probabilistic capacity of general run-length sets in the remaining sections. A detailed discussion and proofs of the theorems can be found in [4].

A. Constrained Systems and Generating Functions

Definition 1. A *constrained system* (\mathcal{A}, w) consists of a countable set \mathcal{A} of strings accepted by the system and an associated weight function $w: \mathcal{A} \rightarrow \mathbb{R}_{>0}$ ($\mathbb{R}_{>0}$ denotes the positive real numbers) with the following property: if $\mathbf{a}, \mathbf{b} \in \mathcal{A}$ and $\text{cat}(\mathbf{a}, \mathbf{b}) \in \mathcal{A}$ then $w[\text{cat}(\mathbf{a}, \mathbf{b})] = w(\mathbf{a}) + w(\mathbf{b})$.

Here and hereafter, $\text{cat}(\mathbf{a}, \mathbf{b})$ denotes the concatenation of \mathbf{a} and \mathbf{b} .

Definition 2. Let (\mathcal{A}, w) represent a constrained system. We define the *generating function* of \mathcal{A} by

$$G_{\mathcal{A}}(s) := \sum_{\mathbf{a} \in \mathcal{A}} e^{-w(\mathbf{a})s}, \quad s \in \mathbb{C} \quad (1)$$

where \mathbb{C} denotes the set of complex numbers.

Let $\Omega = \{w(\mathbf{a}) | \mathbf{a} \in \mathcal{A}\}$ denote the set of distinct string weights of elements in \mathcal{A} . We order and index the set Ω such that $\Omega = \{\nu_i\}_{i=1}^{\infty}$ with $\nu_1 < \nu_2 < \dots$. For every $\nu_i \in \Omega$, $N(\nu_i)$ denotes the number of distinct strings of weight ν_i in \mathcal{A} . A compact representation of the sequence $\{N(\nu_i)\}_{i=1}^{\infty}$ can be obtained by rewriting the generating function as

$$G_{\mathcal{A}}(s) = \sum_{i=1}^{\infty} N(\nu_i) e^{-\nu_i s}. \quad (2)$$

The growth of the sequence $\{N(\nu_i)\}_{i=1}^{\infty}$ is determined by the analytic properties of $G_{\mathcal{A}}$ [5]. This property lies at the heart of the results presented in this section.

B. Combinatorial Capacity

From an information-theoretic perspective, of main interest is the *exponential* growth of the sequence $\{N(\nu_i)\}_{i=1}^{\infty}$, which is captured by the notion of “combinatorial capacity”.

Definition 3. We define the *combinatorial capacity* by

$$C := \limsup_{k \rightarrow \infty} \frac{\ln \left[\sum_{i=1}^k N(\nu_i) \right]}{\nu_k} \quad (3)$$

Here and hereafter, \ln denotes the natural logarithm. Shannon’s original definition of combinatorial capacity was

$$C_0 := \limsup_{i \rightarrow \infty} \frac{\ln [N(\nu_i)]}{\nu_i}. \quad (4)$$

The original definition is only meaningful when the set of distinct string weights Ω is *not too dense*, that is, when there

exists some constant $L \geq 0$ and some constant $K \geq 0$ such that for any integer $n \geq 0$

$$\max_{\nu_k < n} k \leq Ln^K. \quad (5)$$

See [2] and [6] for detailed discussions of the “not too dense” property. The following theorem shows how the combinatorial capacity of a constrained system is related to its generating function and it further shows that our definition of combinatorial capacity is consistent with the original one.

Theorem 1. Let (\mathcal{A}, w) be a constrained system with the generating function $G_{\mathcal{A}}(s)$. The following holds:

- 1) The combinatorial capacity C is equal to the abscissa of convergence Q of $G_{\mathcal{A}}$, i.e., $C = Q$.
- 2) If the set of distinct string weights Ω is not too dense, then $C_0 = Q$ and in particular $C_0 = C$.

C. Entropy Rate of Input Processes

In consistency with [1, Appendix 4], we define the entropy rate of weighted random processes.

Definition 4. The *entropy rate* of a random process $Y = \{Y_i\}_{i=1}^{\infty}$, $Y_i \in \mathcal{Y}$ with an associated weight function $w: \mathcal{Y} \rightarrow \mathbb{R}_{>0}$ is defined as

$$\bar{H}(Y) := \limsup_{k \rightarrow \infty} \frac{H(Y_1, \dots, Y_k)}{E[w(Y_1) + \dots + w(Y_k)]}. \quad (6)$$

The operators E, H are defined as follows: For a discrete random variable X and a deterministic function f , $E[f(X)]$ denotes the expected value of $f(X)$ with respect to the probability mass function (PMF) of X , and $H(X) := E[-\ln(X)]$ denotes the entropy of X .

Let $Y = \{Y_i\}_{i=1}^{\infty}$ denote a random process that generates input for a constrained system. Transmitted over the system are the strings $\text{cat}(Y_1, \dots, Y_k)$. To ensure that the entropy rate as defined in Definition 4 actually reflects the entropy rate of the transmitted strings, we have to ensure that Y generates the strings unambiguously, see [4, Section V]. The following definition guarantees this.

Definition 5. Let $Y = \{Y_i\}_{i=1}^{\infty}$, $Y_i \in \mathcal{Y}$ be a random process and let p_k denote the PMF of (Y_1, \dots, Y_k) . Define the sequence of auxiliary random variables $X_k = \text{cat}(Y_1, \dots, Y_k)$ with the supports truncated to

$$\mathcal{X}_k = \{\text{cat}(y_1, \dots, y_k) | (y_1, \dots, y_k) \in \mathcal{Y}^k: p_k(y_1, \dots, y_k) > 0\}$$

where \mathcal{Y}^k denotes the Cartesian product of k copies of \mathcal{Y} . The process Y is an *input process* of the constrained system (\mathcal{A}, w) if the supports \mathcal{X}_k fulfill both of the following conditions:

- 1) $\bigcup_{k=1}^{\infty} \mathcal{X}_k \subseteq \mathcal{A}$.
- 2) if $j \neq k$, then $\mathcal{X}_j \cap \mathcal{X}_k = \emptyset$.

Condition 1) ensures that Y generates valid strings and condition 2) ensures that Y does so unambiguously. For a discussion of these conditions in the context of general run-length sets, see Subsection VI-A in this work. The entropy rate of an input process of a constrained system relates to the combinatorial capacity as follows.

Theorem 2. Let (\mathcal{A}, w) denote a constrained system. The entropy rate of an input process Y of (\mathcal{A}, w) is upper-bounded by the abscissa of convergence Q of $G_{\mathcal{A}}$, and, in particular, it is upper bounded by the combinatorial capacity C of (\mathcal{A}, w) .

Having defined entropy rate in Definition 4 and input processes in Definition 5, we implicitly defined the probabilistic capacity of a constrained system: it is given by the maximum entropy rate an input process can have.

III. SETUP

A. Definition of $\langle \mathcal{W}, \mathcal{L} \rangle$

The class of constrained systems that we consider in this work can be specified by a set \mathcal{W} of *run-lengths* and a set \mathcal{L} of *labels*. The set of run-lengths \mathcal{W} is a non-empty, countable subset of the positive real numbers $\mathbb{R}_{>0}$. The set of labels \mathcal{L} is nonempty and finite. One *run* is the substring of a string during which the label does not change. We refer to the length of a run r by its weight $w(r)$ and we refer to the label of a run r by the label function $l(r)$. The set of allowed strings of such a system is given by

$$\begin{aligned} \langle \mathcal{W}, \mathcal{L} \rangle := \{ & \text{cat}(r_1, \dots, r_k) \mid k \in \mathbb{N}, \\ & \text{for } i = 1, \dots, k: w(r_i) \in \mathcal{W} \text{ and } l(r_i) \in \mathcal{L}, \\ & \text{for } i = 1, \dots, k-1: l(r_{i+1}) \neq l(r_i) \} \end{aligned} \quad (7)$$

where $\mathbb{N} = \{1, 2, \dots\}$ denotes the natural numbers. From this definition, we see that if the cardinality of \mathcal{L} is equal to one, then each string in $\langle \mathcal{W}, \mathcal{L} \rangle$ consists of only one run with its length in \mathcal{W} and with its label equal to the unique label from \mathcal{L} . From now on, we therefore assume $|\mathcal{L}| \geq 2$. We do *not* require that the set $\langle \mathcal{W}, \mathcal{L} \rangle$ fulfills the “not too dense” property (5). However, we assume that the limit in (3) exists, i.e., that $\langle \mathcal{W}, \mathcal{L} \rangle$ has a well-defined combinatorial capacity.

B. $\langle \mathcal{W}, \mathcal{L} \rangle$ is in General not Regular

It is important to note that $\langle \mathcal{W}, \mathcal{L} \rangle$, in general, does not form a regular language. If it would, it could be analyzed within the framework of finite state machines as defined in [2]. Consider as an example $\langle \mathcal{V}, \mathcal{K} \rangle$ where

$$\mathcal{V} = \{2^k \mid k \in \mathbb{N}\} \quad (8)$$

$$\mathcal{K} = \{0, 1\}. \quad (9)$$

This is for $\mathcal{A} = \mathbb{N}$ and $\xi = 2$ an example for the asynchronous $[\mathcal{A}, \xi]$ channel with binary input as introduced in [3].

The first way to argue that $\langle \mathcal{V}, \mathcal{K} \rangle$ is not regular is to interpret the set of runs

$$\{r \mid w(r) \in \mathcal{V}, l(r) \in \mathcal{K}\} \quad (10)$$

as an infinite alphabet. The set $\langle \mathcal{V}, \mathcal{K} \rangle$ is thus generated by an infinite alphabet, whereas a regular language is by definition generated by a finite alphabet [7]. However, the set $\langle \mathcal{V}, \mathcal{K} \rangle$ can also be generated by concatenating the two runs r_0, r_1 with $w(r_0) = w(r_1) = 1$ and $l(r_0) = 0, l(r_1) = 1$. Indeed we now have a finite alphabet $\{r_0, r_1\}$ by which we can generate $\langle \mathcal{V}, \mathcal{K} \rangle$, but we need an infinite memory: only the

concatenation of $1, 2, 4, 8, \dots$ runs of the same label result in a valid run-length. This memory cannot be implemented by a finite-state machine, which again shows that $\langle \mathcal{V}, \mathcal{K} \rangle$ is not regular.

C. How to Derive the Probabilistic Capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$

Our aim is to derive the probabilistic capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$ in the general case, that is, we want to derive the maximum entropy rate an input process of $\langle \mathcal{W}, \mathcal{L} \rangle$ can have. According to Theorem 2, the maximum entropy rate is upper bounded by the combinatorial capacity, which itself is, by Theorem 1, given by the abscissa of convergence of the generating function. Our approach is now as follows:

- 1) We first derive the generating function.
- 2) We then use the generating function to calculate the combinatorial capacity.
- 3) We finally define an input process whose entropy rate is equal to the combinatorial capacity.

After accomplishing these three tasks, we have shown that the probabilistic capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$ is equal to its combinatorial capacity and that its value can be derived by using the corresponding formulas for the combinatorial capacity. It should be noted that this approach is also suitable for other types of constrained systems. For instance, it was used in [4] to derive the combinatorial and the probabilistic capacity of (j, k) constraints.

IV. GENERATING FUNCTION OF $\langle \mathcal{W}, \mathcal{L} \rangle$

We start by deriving the generating function of $\langle \mathcal{W}, \mathcal{L} \rangle$. We do this run by run. The generating function of the run-length of the first run is given by

$$G_{\mathcal{W}}(s) = \sum_{\nu \in \mathcal{W}} e^{-\nu s}. \quad (11)$$

For the first run, we can choose from among $|\mathcal{L}|$ labels, so the generating function of the first run R_1 is given by

$$G_{R_1}(s) = |\mathcal{L}| G_{\mathcal{W}}(s). \quad (12)$$

The generating function of the run-lengths of the second run is again $G_{\mathcal{W}}(s)$, however, given the label chosen for the first run, we can for the second run only choose from among $|\mathcal{L}| - 1$ labels. The first two runs result from concatenating the first run with the second run. Since we guarantee that two subsequent runs have different labels, concatenating the first with the second run corresponds to multiplying the corresponding generating functions [8, Chapter 2]. The generating function of the first two runs is thus

$$G_{R_2}(s) = |\mathcal{L}| G_{\mathcal{W}}(s) (|\mathcal{L}| - 1) G_{\mathcal{W}}(s). \quad (13)$$

The same as for the second run applies for all subsequent runs, so the generating function of the first k runs is given by

$$G_{R_k}(s) = |\mathcal{L}| G_{\mathcal{W}}(s) [(|\mathcal{L}| - 1) G_{\mathcal{W}}(s)]^{k-1} \quad (14)$$

To get the complete generating function of $\langle \mathcal{W}, \mathcal{L} \rangle$ we have to add up all G_{R_k} .

$$G_{\langle \mathcal{W}, \mathcal{L} \rangle}(s) = \sum_{k=1}^{\infty} G_{R_k}(s) \quad (15)$$

$$= |\mathcal{L}| G_{\mathcal{W}}(s) \sum_{m=0}^{\infty} [(|\mathcal{L}| - 1) G_{\mathcal{W}}(s)]^m. \quad (16)$$

V. COMBINATORIAL CAPACITY OF $\langle \mathcal{W}, \mathcal{L} \rangle$

With the help of the generating function (16) as derived in the previous section, we can now derive the combinatorial capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$ by applying Theorem 1. Let $\text{Re}(s)$ denote the real part of s . From [9, Theorem 3], we know that $G_{\langle \mathcal{W}, \mathcal{L} \rangle}(s)$ converges if and only if $G_{\langle \mathcal{W}, \mathcal{L} \rangle}[\text{Re}(s)]$ converges, and since $G_{\langle \mathcal{W}, \mathcal{L} \rangle}$ is strictly positive on the real axis, the latter converges if and only if

$$(|\mathcal{L}| - 1) G_{\mathcal{W}}[\text{Re}(s)] < 1. \quad (17)$$

Thus, as a corollary to Theorem 1, we have

Proposition 1. *The combinatorial capacity C of $\langle \mathcal{W}, \mathcal{L} \rangle$ is given by the unique positive real solution of*

$$(|\mathcal{L}| - 1) G_{\mathcal{W}}(s) = 1. \quad (18)$$

This proposition coincides with [3, Theorem 2]. As observed in [3], for $|\mathcal{L}| = 2$, the proposition reduces to [10, Proposition 1.1]. Furthermore, if $\mathcal{L} = \{0, 1\}$ and $\mathcal{W} = \{1, 2, \dots, k\}$, then the proposition provides the combinatorial capacity of the $(j = k, k)$ run-length constraint and coincides with the formulas derived in [4], [11].

VI. PROBABILISTIC CAPACITY OF $\langle \mathcal{W}, \mathcal{L} \rangle$

To calculate the probabilistic capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$, it remains to define an input process for $\langle \mathcal{W}, \mathcal{L} \rangle$ that has an entropy rate equal to the combinatorial capacity C .

A. Preliminary Considerations

Not every process $Y = \{Y_i\}_{i=1}^{\infty}$ with $Y_i \in \langle \mathcal{W}, \mathcal{L} \rangle$ is a valid input process. The first reason is that a concatenation of elements from $\langle \mathcal{W}, \mathcal{L} \rangle$ is not necessarily an element of $\langle \mathcal{W}, \mathcal{L} \rangle$. Let, for instance, $\langle \mathcal{W}, \mathcal{L} \rangle$ be given by

$$\mathcal{W} := \{1, 2, \pi\}, \quad \mathcal{L} := \{\text{red}, \text{green}\}. \quad (19)$$

Define the run r by $w(r) = \pi$ and $l(r) = \text{red}$. Run r is an element of $\langle \mathcal{W}, \mathcal{L} \rangle$. The string $\text{cat}(r, r)$ is of color red, therefore, $\text{cat}(r, r)$ is just one run. But $w[\text{cat}(r, r)] = 2\pi \notin \mathcal{W}$, so the concatenation of r with r is not in $\langle \mathcal{W}, \mathcal{L} \rangle$ and Y violates condition 1) of Definition 5.

A second reason why Y may not be an input process is ambiguity: define r_1 by $w(r_1) = 1$ and $l(r_1) = \text{red}$ and define r_2 by $w(r_2) = 2$ and $l(r_2) = \text{red}$. Obviously, $r_1, r_2 \in \langle \mathcal{W}, \mathcal{L} \rangle$. But $\text{cat}(r_1, r_1) = r_2$, so r_2 can be a realization either of (Y_1, Y_2) or just of Y_1 . This violates condition 2) of Definition 5.

We leave the example and let from now on $\langle \mathcal{W}, \mathcal{L} \rangle$ again be an arbitrary general run-length set with $|\mathcal{L}| \geq 2$ and $|\mathcal{W}|$

possibly countably-infinite. We choose the following approach to construct an input process that guarantees both condition 1) and condition 2) of Definition 5: we define a subset \mathcal{Y} of $\langle \mathcal{W}, \mathcal{L} \rangle$ such that if the Y_i take values in \mathcal{Y} , then both conditions from Definition 5 are automatically fulfilled.

B. The Support of a Class of Input Processes of $\langle \mathcal{W}, \mathcal{L} \rangle$

We choose an arbitrary but fixed element $l_0 \in \mathcal{L}$ and consider the subset \mathcal{Y} of $\langle \mathcal{W}, \mathcal{L} \rangle$ that consists of strings where the first run has label l_0 and where all subsequent runs have a label different from l_0 . Note that each string in \mathcal{Y} consists of at least two runs. The set \mathcal{Y} has two important properties:

- 1) If $x \in \mathcal{Y}$ and $y \in \mathcal{Y}$ then $\text{cat}(x, y) \in \langle \mathcal{W}, \mathcal{L} \rangle$, so

$$\bigcup_{i=1}^{\infty} \{\text{cat}(x_1, \dots, x_i) | (x_1, \dots, x_i) \in \mathcal{Y}^i\} \subseteq \langle \mathcal{W}, \mathcal{L} \rangle.$$

- 2) Each string in $\langle \mathcal{W}, \mathcal{L} \rangle$ that starts with a run of label l_0 and ends with a run of a label different from l_0 can be unambiguously generated by concatenating elements from \mathcal{Y} .

In other words, any random process $\{Y_i\}_{i=1}^{\infty}$, $Y_i \in \mathcal{Y}$ fulfills condition 1) and condition 2) of Definition 5 and is therefore an input process of $\langle \mathcal{W}, \mathcal{L} \rangle$.

In the following derivations, we will also need the generating function of \mathcal{Y} . It is given by

$$G_{\mathcal{Y}}(s) = \underbrace{G_{\mathcal{W}}(s)}_{\text{one run with label } l_0} \underbrace{(|\mathcal{L}| - 1) G_{\mathcal{W}}(s)}_{\text{one run with label from } \mathcal{L} \setminus l_0} \sum_{k=0}^{\infty} \underbrace{\left[(|\mathcal{L}| - 2) G_{\mathcal{W}}(s) \right]^k}_{\text{0 or more runs with labels from } \mathcal{L} \setminus l_0 \text{ but different from the label of the 2nd run}}. \quad (20)$$

The derivation of $G_{\mathcal{Y}}$ follows the lines of Section IV.

C. A Maxentropic Input Process of $\langle \mathcal{W}, \mathcal{L} \rangle$

For a random process that takes values in \mathcal{Y} as defined in the previous subsection, we now want to define a PMF in such a way that the entropy rate of the resulting input process is equal to the combinatorial capacity C and thereby maxentropic. To this end, we will need the following proposition:

Proposition 2. *Let $Z = \{Z_i\}_{i=1}^{\infty}$ denote a random process with each Z_i taking values in the countably infinite set \mathcal{Z} and let $w: \mathcal{Z} \rightarrow \mathbb{R}_{>0}$ be an associated positive weight function. Let R denote the unique positive real solution of*

$$G_{\mathcal{Z}}(s) = 1 \quad (21)$$

and define $p_{\mathcal{Z}}$ as

$$p_{\mathcal{Z}}(z) := e^{-w(z)R}, \quad z \in \mathcal{Z}. \quad (22)$$

Then $p_{\mathcal{Z}}$ is a PMF and by letting the Z_i be independent and identically distributed (IID) according to $p_{\mathcal{Z}}$,

$$\bar{H}(Z) = R. \quad (23)$$

Proof: For a finite set \mathcal{Z} , this proposition is equivalent to [12, Theorem 1]. By [4, Lemma 1] from our work, the generalization to countably infinite \mathcal{Z} follows directly. ■

Let $Y = \{Y_i\}_{i=1}^\infty$ denote a random process where the Y_i take values in \mathcal{Y} , with \mathcal{Y} as defined in the previous subsection. We first show that $G_Y(C) = 1$. Recall from (18) that

$$(|\mathcal{L}| - 1) G_W(C) = 1. \quad (24)$$

We thus have

$$G_Y(C) = G_W(C) \underbrace{(|\mathcal{L}| - 1) G_W(C)}_{=1} \sum_{k=0}^{\infty} \underbrace{\left[(|\mathcal{L}| - 2) G_W(C) \right]^k}_{=1 - G_W(C)} \quad (25)$$

$$= G_W(C) \sum_{k=0}^{\infty} [1 - G_W(C)]^k \quad (26)$$

$$= \frac{G_W(C)}{1 - [1 - G_W(C)]} \quad (27)$$

$$= \frac{G_W(C)}{G_W(C)} = 1 \quad (28)$$

where (25) follows from (20), (26) follows from (24), and where we used the geometric series formula in (27). By Proposition 2,

$$p_Y(y) := e^{-w(y)C}, \quad y \in \mathcal{Y} \quad (29)$$

is a PMF. We let the Y_i be IID according to p_Y . As a consequence, again by Proposition 2,

$$\bar{H}(Y) = C \quad (30)$$

Thus, the entropy rate of Y is equal to the combinatorial capacity C , and since C is according to Theorem 2 an upper bound, Y is maxentropic. We thus have the following proposition.

Proposition 3. *The probabilistic capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$ is equal to its combinatorial capacity.*

D. Discussion of [3]

1) *Definition of probabilistic capacity:* Proposition 3 was proved in [3] with respect to the following definition of probabilistic capacity. Let $Y = \{Y_i\}_{i=1}^\infty$ denote an input process of $\langle \mathcal{W}, \mathcal{L} \rangle$. Let Y generate strings until time instant T . If the last run is not complete, discard the incomplete last run. Let ν_0 denote the smallest element from \mathcal{W} . The strings that Y can generate in this way are for each time instant T given by the set

$$\mathcal{X}_T = \{s \in \langle \mathcal{W}, \mathcal{L} \rangle | w(s) \leq T, w(s) + \nu_0 > T\}. \quad (31)$$

Define for each time instant T an auxiliary random variable X_T with the support \mathcal{X}_T . Since the input process Y generates one of the strings from \mathcal{X}_T until time instant T with probability 1, the PMF of Y implies a PMF of X_T . The authors of [3] now define the entropy rate of Y by

$$\bar{H}(Y) := \limsup_{T \rightarrow \infty} \frac{H(X_T)}{T} \quad (32)$$

and define the probabilistic capacity of $\langle \mathcal{W}, \mathcal{L} \rangle$ by the maximum of \bar{H} over all input processes Y . With respect to this definition, it is easy to show Theorem 2, but it is difficult

to show Proposition 3. This observation is reflected by the length of the proof of [3, Theorem 4] given in [3]. As a side-result of our work, both definitions of probabilistic capacity are equivalent, but their calculations differ.

2) *Maxentropic input process:* In [3], the authors solve the problem of violating the conditions from Definition 5 by defining an input process with memory, specifically, a Markov chain with $|\mathcal{L}|$ states. Basically, they consider PMFs of Y where

$$\text{Prob}[l(Y_{i+1}) = \mathbf{t} | l(Y_i) = \mathbf{t}] = 0, \quad \forall \mathbf{t} \in \mathcal{L} \quad (33)$$

i.e., two subsequent runs generated by the process have the same label with probability zero. It can be shown that the entropy rate as defined in Definition 4 of this process is equal to the combinatorial capacity C , i.e., this process is also maxentropic with respect to our definition of entropy rate.

VII. CONCLUSIONS

In this work, we used generating functions to show in a unified manner that the probabilistic capacity of a general run-length set is equal to its combinatorial capacity. This is an interesting result, since general run-length sets are in general not regular and can not be analyzed within the framework of finite state machines, which has been the usual approach for the analysis of constrained systems so far. Generating functions and their information-theoretic properties, which we recently established, appear to be a strong tool for the analysis of a broad class of constrained systems. We believe that known results can be re-established within this new framework in a unified manner and that new results can be obtained. The present work is a first step in this direction.

REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, Jul. and Oct. 1948.
- [2] A. Khandekar, R. McEliece, and E. Rodemich, "The discrete noiseless channel revisited," in *Coding, Communications, and Broadcasting*. Research Studies Press Ltd., 2000, pp. 115–137.
- [3] R. Yeung, N. Cai, S.-W. Ho, and A. Wagner, "Reliable communication in the absence of a common clock," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 700–712, 2009.
- [4] G. Böcherer, V. C. da Rocha Jr., C. Pimentel, and R. Mathar, "On the capacity of constrained systems," in *Proc. Int. ITG Conf. Source Channel Coding*, 2010. [Online]. Available: <http://arxiv.org/abs/0911.1090>
- [5] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge University Press, 2008.
- [6] G. Böcherer, V. C. da Rocha Jr., and C. Pimentel, "Capacity of general discrete noiseless channels," in *Proc. Int. Symp. Commun. Applicat. (ISCTA)*, 2007. [Online]. Available: <http://arxiv.org/abs/0802.2451>
- [7] M. Sipser, *Introduction to the Theory of Computation*, 2nd ed. Thomson Course Technology, 2006.
- [8] G. Böcherer, "Analytic asymptotics of discrete noiseless channels," Master's thesis, ETH Zurich, 2007. [Online]. Available: <http://www.georg-boecherer.de/repository/analyticAsymptotics.pdf>
- [9] G. H. Hardy and M. Riesz, *The General Theory of Dirichlet's Series*. Cambridge: at the University Press, 1915.
- [10] I. Csiszár, "Simple proofs of theorems on noiseless channels," *Inf. Contr.*, vol. 14, pp. 285–298, 1969.
- [11] T. Mittelholzer, "Enumerative maximum-transition-run codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2009, pp. 1549–1553.
- [12] R. M. Krause, "Channels which transmit letters of unequal duration," *Inf. Contr.*, vol. 5, pp. 3–24, 1962.